

LAMPIRAN

RANCANG BANGUN SISTEM MONITORING PARAMETER PLTS BERPENDINGIN UDARA DENGAN APLIKASI TELEGRAM BERBASIS NODEMCU ESP8266

ORIGINALITY REPORT

14%

SIMILARITY INDEX

14%

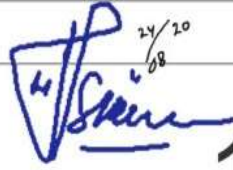
INTERNET SOURCES

1%

PUBLICATIONS

6%

STUDENT PAPERS



PRIMARY SOURCES

1	eprints.uny.ac.id Internet Source	4%
2	media.neliti.com Internet Source	1%
3	repository.its.ac.id Internet Source	1%
4	eprints.umg.ac.id Internet Source	1%
5	jurnal.umk.ac.id Internet Source	1%
6	Submitted to Sogang University Student Paper	1%
7	digilib.unila.ac.id Internet Source	1%
8	docplayer.info Internet Source	1%

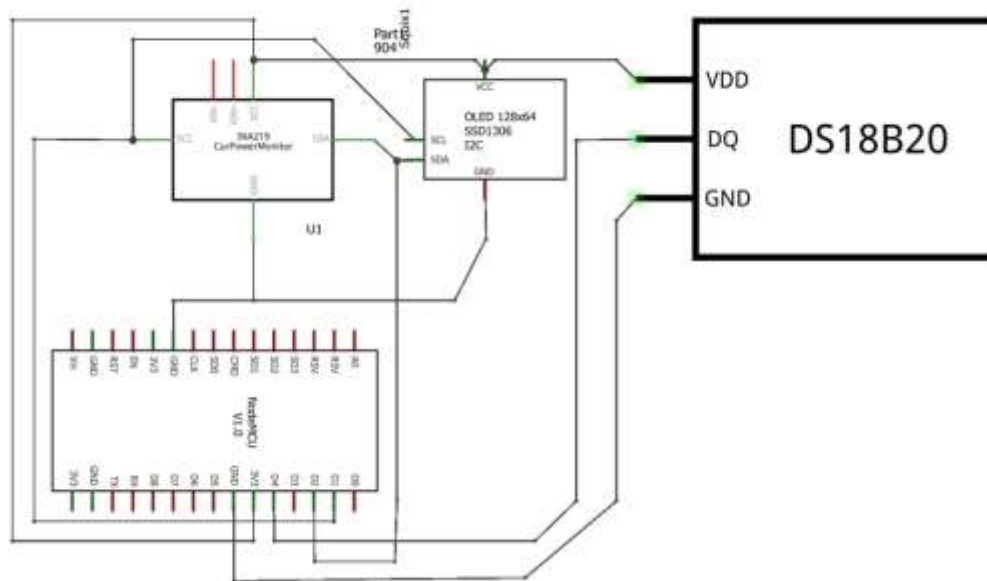
9	widuri.raharja.info Internet Source	1%
10	lib.unnes.ac.id Internet Source	1%
11	Submitted to Sultan Agung Islamic University Student Paper	1%
12	es.scribd.com Internet Source	1%

Exclude quotes Off

Exclude matches < 1%

Exclude bibliography Off

Lampiran 1. Wirring Rangkaian Keseluruhan



Lampiran 2. Program NodeMCU ESP8266

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <DallasTemperature.h>
#include <OneWire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Sensor.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define ONE_WIRE_BUS 2 //D1 pin of nodemcu

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire); // Pass the
oneWire reference to Dallas Temperature.
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

const int INA_addr = 0x40; // INA219 address

Adafruit_INA219 ina219(INA_addr);

WiFiClient client;
WiFiClientSecure client1;

#define BOTtoken "1039381693:AAGTMtEDMlt3MlIYDyLV3_3lsTi6VOifUYs"
#define idChat "1061495775" //idbot
String thingSpeakAddress = "http://api.thingspeak.com";
String apiKey = "VQIMG49347JT77Z6";
float t = 0;
float busVoltage = 0;
float current = 0; // Measure in milli amps
float power = 0;
String myData;
const char *ssid = "tekon_password_e"; // replace with your
wifi ssid and wpa2 key
const char *pass = "12345678";

UniversalTelegramBot bot(BOTtoken, client1);

int Bot_mtbs = 1000; //mean time between scan messages
long Bot_lasttime; //last time messages' scan has been done
bool Start = false;

void setup()
{
  bot._debug = true;
  Serial.begin(115200);
  delay(10);
  sensors.begin();
  ina219.begin();
```

```

uint32_t currentFrequency;

if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println(F("SSD1306 allocation failed"));
  for(;;);
}
Wire.setClock(400000L); //set I2C clock to 400kHz

display.clearDisplay();
display.setTextColor(WHITE, BLACK);
display.setTextSize(1);
display.setCursor(0, 0);
display.println(" Monitoring PLTS");
display.display();

WiFi.begin(ssid, pass);

Serial.print("Connecting to ");
display.setCursor(0, 24);
display.println("Connecting...");
display.display();
Serial.println(ssid);
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
display.print("WiFi connected");
display.display();
client1.setInsecure();
delay(2000);
display.clearDisplay();
display.setTextColor(WHITE);
display.display();
}

void handleNewMessages(int numNewMessages) {
  Serial.println("handleNewMessages");
  Serial.println(String(numNewMessages));

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    String text = bot.messages[i].text;

    String from_name = bot.messages[i].from_name;
    if (from_name == "") from_name = "Guest";

    if (text == "Cek arus"){
      myData = "Arus : ";
      myData += current;
      myData += " mA";
      bot.sendMessage(chat_id, myData, "");
    }
    if (text == "Cek tegangan"){

```

```

        myData = "Tegangan : ";
        myData += busVoltage;
        myData += "V";
        bot.sendMessage(chat_id, myData, "");
    }

    if (text == "Cek suhu") {
        myData = "Suhu :";
        myData += t;
        myData += "C";
        bot.sendMessage(chat_id, myData, "");
    }

    if (text == "Cek daya") {
        myData = "Daya :";
        myData += power;
        myData += " W";
        bot.sendMessage(chat_id, myData, "");
    }

    if (text == "Cek semua") {
        myData += "Monitoring PLTS Berpendingin Udara.\n\n";
        myData = "Suhu :";
        myData += t;
        myData += " C";
        myData += "\nTegangan :";
        myData += busVoltage;
        myData += " V";
        myData += "\nArus : ";
        myData += current;
        myData += "mA";
        myData += "\nDaya : ";
        myData += power;
        myData += "W";
        bot.sendMessage(chat_id, myData, "");
    }

    if (text == "Start") {
        String welcome = "Welcome to Universal Arduino Telegram Bot
library, " + from_name + ".\n";
        welcome += "Monitoring PLTS Berpendingin Udara.\n\n";
        welcome += "Cek arus : untuk melihat data arus\n";
        welcome += "Cek tegangan : untuk melihat data tegangan\n";
        welcome += "Cek suhu : untuk melihat data suhu\n";
        welcome += "Cek Daya :untuk melihat data daya keluaran \n ";
        welcome += "Cek Semua : untuk mengecek semuanya\n";
        bot.sendMessage(chat_id, welcome, "Markdown");
    }
}
}

void loop()
{
    delay(2000);
    sensors.requestTemperatures();
    // Read temperature as Fahrenheit (isFahrenheit = true)

```

```

t = sensors.getTempCByIndex(0);
busVoltage = ina219.getBusVoltage_V();
current = ina219.getCurrent_mA();
power = busVoltage * (current/1000); // Calculate the Power

 kirim_thingspeak(t, power, busVoltage, current);

 /*if (isnan(power) || isnan(t) || isnan(busVoltage) ||
isnan(current)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
}*/

if (millis() > Bot_lasttime + Bot_mtbs) {
    int numNewMessages = bot.getUpdates(bot.last_message_received
+ 1);

    while (numNewMessages) {
        Serial.println("got response");
        handleNewMessages(numNewMessages);
        numNewMessages = bot.getUpdates(bot.last_message_received +
1);
    }

    Bot_lasttime = millis();
}
if (t > 50.00) {
    bot.sendChatAction(idChat, "Sedang mengetik...");
    Serial.print("Suhu saat ini : ");
    Serial.println(t);
    delay(3000);
    String suhu = "Intensitas suhu : ";
    suhu += int(t);
    suhu += " *C\n";
    suhu += "Suhu PLTS Meningkatkan!\n";
    bot.sendMessage(idChat, suhu, "");
    Serial.print("Mengirim data sensor ke telegram");
}

if (current < 0.0) {
    bot.sendChatAction(idChat, "Sedang mengetik...");
    Serial.print("arus saat ini : ");
    Serial.println(current);
    delay(3000);
    String arus = "Arus : ";
    arus += int(current);
    arus += " mA\n";
    arus += "Arus PLTS Menurun!\n";
    bot.sendMessage(idChat, arus, "");
    Serial.print("Mengirim data sensor ke telegram");
}

if (busVoltage < 0.9) {
    bot.sendChatAction(idChat, "Sedang mengetik...");
    Serial.print("Tegangan saat ini : ");

```



```

    Serial.println(busVoltage);
    delay(3000);
    String tegangan = "Tegangan : ";
    tegangan += int(busVoltage);
    tegangan += " V\n";
    tegangan += "Tegangan PLTS Menurun!\n";
    bot.sendMessage(idChat, tegangan, "");
    Serial.print("Mengirim data sensor ke telegram");
}
display.clearDisplay();

// display temperature
display.setTextSize(1);
display.setCursor(0,0);
display.print("T: ");
display.setTextSize(1);
display.setCursor(20,0);
display.print(t);
display.print(" ");
display.setTextSize(1);
display.cp437(true);
display.write(167);
display.setTextSize(1);
display.print("C");
//display voltage
display.setTextSize(1);
display.setCursor(0, 15);
display.print("V: ");
display.setTextSize(1);
display.setCursor(20, 15);
display.print(busVoltage);
display.print(" V");
//display current
display.setTextSize(1);
display.setCursor(0, 30);
display.print("I: ");
display.setTextSize(1);
display.setCursor(20, 30);
display.print(current);
display.print(" mA");
// display Power
display.setTextSize(1);
display.setCursor(0, 45);
display.print("P: ");
display.setTextSize(1);
display.setCursor(20, 45);
display.print(power);
display.print("W");

display.display();
}

void kirim_thingspeak(float t, float power, float busVoltage,
float current) {
    if (client.connect("api.thingspeak.com", 80)) {

```

```

String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(t);
    postStr += "&field2=";
    postStr += String(busVoltage);
    postStr += "&field3=";
    postStr += String(current);
    postStr += "&field4=";
    postStr += String(power);
    postStr += "\r\n\r\n";

    client.print("POST /update
HTTP/1.1\n");
    client.print("Host:
api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY:
"+apiKey+"\n");
    client.print("Content-Type:
application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);
    Serial.println("%s. Send to
Thingspeak.");
}
unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}
while (client.available()) {
    String line = client.readStringUntil('\r');
}
    //client.stop();

    Serial.println("Waiting...");

    // thingspeak needs minimum 15 sec delay between updates
    delay(1000);
}

```

Lampiran 3. Data Spesifikasi panel surya 20 Wp *Monocrystalline*

No	Data spesifikasi panel surya 20 Wp <i>Monocrystalline</i>	
1	Material sell	Mono
2	Daya Maksimum	20 W
3	Toleransi produk	$\pm 3 \%$
4	Arus Maksimum (I_{mp})	1,12 A
5	Tegangan Maksimum (V_{mp})	18 V
6	Arus saat hubung singkat (I_{sc})	1,23 A
7	Tegangan Saat terhubung (V_{oc})	21,6 V
8	Suhu Maksimum yang disarankan	25°C

Lampiran 4. Data Spesifikasi NodeMCU ESP8266

SPESIFIKASI	NODEMCU V3
Mikrokontroler	ESP8266
Ukuran Board	57 mm x 30 mm
Tegangan Input	3.3 - 5V
GPIO	13 Pin
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4 MB
Clock Speed	40/26/24 MHz
WiFi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz - 2.5 GHz
USB Port	Micro USB
USB to Serial Converter	CH340G

Lampiran 5. Data Spesifikasi sensor INA219

PARAMETER	TEST CONDITIONS	INA219A			INA219B			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
INPUT								
V_{INPUT}	Full-scale current sense (input) voltage range	PGA = /1	0	±40	0	±40	mV	
		PGA = /2	0	±80	0	±80	mV	
		PGA = /4	0	±160	0	±160	mV	
		PGA = /8	0	±320	0	±320	mV	
	Bus voltage (input voltage) range ⁽²⁾	BRNG = 1	0	32	0	32	V	
		BRNG = 0	0	16	0	16	V	
CMRR	Common-mode rejection	$V_{\text{IN}} = 0$ to 26 V	100	120	100	120	dB	
V_{OS}	Offset voltage, RTI ⁽³⁾	PGA = /1	±10	±100	±10	±50 ⁽⁴⁾	µV	
		PGA = /2	±20	±125	±20	±75 ⁽⁴⁾	µV	
		PGA = /4	±30	±150	±30	±75 ⁽⁴⁾	µV	
		PGA = /8	±40	±200	±40	±100 ⁽⁴⁾	µV	
	vs Temperature	$T_A = -25^{\circ}\text{C}$ to 85°C	0.1		0.1	µV/°C		
PSRR	vs Power Supply	$V_B = 3$ to 5.5 V	10		10	µV/V		
	Current sense gain error vs Temperature	$T_A = -25^{\circ}\text{C}$ to 85°C		±40		±40	m%	
	IN+ pin input bias current	Active mode		20		20	µA	
	IN- pin input bias current V_{IN} - pin input impedance	Active mode	20 320		20 320		µA kΩ	
	IN+ pin input leakage ⁽⁵⁾	Power-down mode	0.1	±0.5	0.1	±0.5	µA	
	IN- pin input leakage ⁽⁵⁾	Power-down mode	0.1	±0.5	0.1	±0.5	µA	
DC ACCURACY								
	ADC basic resolution		12		12		bits	
	Shunt voltage, 1 LSB step size		10		10		µV	
	Bus voltage, 1 LSB step size		4		4		mV	
	Current measurement error over Temperature	$T_A = -25^{\circ}\text{C}$ to 85°C	±0.2%	±0.5%	±0.2%	±0.3% ⁽⁴⁾		
			±1%		±0.5% ⁽⁴⁾			
	Bus voltage measurement error over Temperature	$T_A = -25^{\circ}\text{C}$ to 85°C	±0.2%	±0.5%	±0.2%	±0.5%		
			±1%		±1%			
	Differential nonlinearity		±0.1		±0.1		LSB	
ADC TIMING								
	ADC conversion time	12 bit		532	596	532	596	µs
		11 bit		276	304	276	304	µs
		10 bit		148	163	148	163	µs
		9 bit		84	93	84	93	µs
	Minimum convert input low time		4		4		µs	
SMBus								
	SMBus timeout ⁽⁶⁾		28	35	28	35	ms	
DIGITAL INPUTS (SDA as Input, SCL, A0, A1)								
	Input capacitance		3		3		pF	
	Leakage input current	$0 \leq V_{\text{IN}} \leq V_B$	0.1	1	0.1	1	µA	
	V_{IH} input logic level		0.7 (V_B)	6	0.7 (V_B)	6	V	
	V_{IL} input logic level		-0.3	0.3 (V_B)	-0.3	0.3 (V_B)	V	
PARAMETER	TEST CONDITIONS	INA219A			INA219B			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
	Hysteresis		500		500		mV	
OPEN-DRAIN DIGITAL OUTPUTS (SDA)								
	Logic 0 output level	$I_{\text{OL}} = 3$ mA	0.15	0.4	0.15	0.4	V	
	High-level output leakage current	$V_{\text{OUT}} = V_B$	0.1	1	0.1	1	µA	
POWER SUPPLY								
	Operating supply range		3	5.5	3	5.5	V	
	Quiescent current		0.7	1	0.7	1	mA	
	Quiescent current, power-down mode		6	15	6	15	µA	
	Power-on reset threshold		2		2		V	

Lampiran 6. Data Spesifikasi Sensor DS18B20

Power supply	3.0V - 5.5V
Operating temperature range	-55°C to + 125 °C(-67 F to +257F)
Storage temperature range	-55°C to + 125 °C(-67 F to +257F)
Accuracy over the range of - 10 °C to +85 °C	± 0.5 °C
3-pin 2510 female header housing water proof	
Stainless steel sheath	
Size of Sheath	6*50 mm
Connector :	RJ11/RJ12, 3P-2510, USB
Pin Definition	RED : VCC Yellow: DATA Black : GND
Cable Length	1 meter, 2m,3m, 4m are available upon request .